



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1969

A simulation model of the IBM 360 computer
in an MVT environment.

Varnell, Allen Kent

Monterey, California. U.S. Naval Postgraduate School

<http://hdl.handle.net/10945/13041>

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NPS ARCHIVE
1969
VARNELL, A.

A SIMULATION MODEL OF THE IBM 360
COMPUTER IN AN MVT ENVIRONMENT

by

Allen Kent Varnell

INTERNALLY DISTRIBUTED
REPORT

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

UNITED STATES
NAVAL POSTGRADUATE SCHOOL



THESIS

A SIMULATION MODEL OF THE IBM
360 COMPUTER IN AN MVT ENVIRONMENT

by

Allan Kent Varnell

April 1969

*This document has been approved for public re-
lease and sale; its distribution is unlimited.*

LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

INTERNALLY DISTRIBUTED

REPORT

A SIMULATION MODEL OF THE IBM 360

COMPUTER IN AN MVT ENVIRONMENT

by

Allan K. Varnell
Captain, United States Army
B.S., United States Military Academy, 1963

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5101

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN

OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
April 1969

PS + KCH/V327 C.1
269
ARNELL, A.

ABSTRACT

This paper reports on the design of a computer simulation model written in GPSS. A brief description of the System/360 operating system is given. The model consists of macroscopic modules representing distinguishable computer tasks which are capable of independent operation and/or more detailed expansion. A pseudo jobstream of sufficient detail was used to test the viability of this model. Guidelines for experimentation (which was prohibited by a complete lack of data) are outlined; suggested uses of the model are given.

TABLE OF CONTENTS

Chapter		Page
I	INTRODUCTION	7
	A. Background	7
	B. Definition of the Problem	8
II	SYSTEM DESCRIPTION	10
III	MODEL DESIGN	12
	A. GENERAL	12
	B. JOBSTREAM GENERATOR.	13
	C. READER/INTERPRETER	14
	D. OUTPUT WRITER	15
	E. IPL	16
	F. INITIATOR/TERMINATOR	16
	G. INTERRUPTS	18
	H. I/O	18
	I. PRIORITY	19
IV	RESOLUTION	20
	A. RESULTS	20
	B. RECOMMENDATIONS	21
	C. CONCLUDING REMARKS	22
V	SUMMARY	24
	BIBLIOGRAPHY	26
	APPENDIX A - SYSTEM FLOW CHARTS	27
	APPENDIX B - MODEL (GPSS/360)	29
	APPENDIX C - MODEL FLOW CHARTS	35

LIST OF TABLES

Table		Page
1	Parameter Association for Jobstream Generator	13
2	Parameter values for Pseudo Jobstream	14
3	Channel and Control Unit Calculation Formulas	19
4	Sample Results from Pseudo Jobstream Runs	20

ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisor, Mr. G. L. Barksdale, Jr. for his patient help both during and after the composition of this paper.

To the staff of the NPGS Computer Facility my heartfelt thanks for putting up with my always being under foot and processing countless jobs in support of this effort.

Most of my gratitude, however, belongs to my wife, Janet, who suffered through many nights alone while I was fighting the computer and who unselfishly provided typing support for draft after draft.

CHAPTER I

I. INTRODUCTION

A well designed simulation model can effectively predict system performance under a variety of conditions for a relatively low cost. This efficiency becomes an even more important tool when the system is an expensive computer where even subtle changes are costly in time and money. Such a model has been constructed for the IBM 360/67-2 recently acquired by the U.S. Naval Postgraduate School: it is basically macroscopic in design, although facilities have been provided throughout for refinement of statistics at the microscopic level. It depicts operation utilizing the IBM Operating System/360 in an MVT environment. The model developed allows one to analyze an existing system or test proposed configurations.

A. BACKGROUND

Since the arrival at the school of the System/360, the system implementation and operation has been based on the recommendations provided by IBM engineers and literature, and those of the school's staff of system programmers. Although the operation has continually improved over the past eighteen months, it has been at no small expense to the users; system configuration and development experiments have been performed with the normal jobstream.

The users consist of students, staff and faculty in varying percentages, dependent on the time of year. Student usage consists of beginning FORTRAN programmers, Computer Science students engaged in advanced and systems programming, and thesis students. The staff work

is devoted to maintenance of academic records and schedules, and such operations as the student text and classified document libraries. Faculty utilization is by computer class instructors and those engaged in research projects. The computer staff's applications and systems programmers are, of course, utilizing the facility. With few exceptions, all users are in equal competition for the use of the computer, and at no expense to the individual except for his time. It is not unusual to find the backlog of jobs waiting to be processed in excess of twenty-four hours at the peak period of the quarter.

Although work is now in progress on a comprehensive accounting system, timing data on the jobstream is not presently available. The information needed to properly design the priority scheduling algorithm, necessary for efficient machine utilization, is also unavailable.

Except for WATFOR batch runs and a few other jobs handled on an individual basis, the jobs are run on a first-in-first-out basis. This results in lengthy turnaround, much operator time spent in the mounting and dismounting of special disk packs, and an occasional system lockup when a job with unusually large requirements unexpectedly enters what might already be an overtasked system environment.

The possibility of the addition of equipment (core and direct access facilities) to improve overall performance and the reconfiguration or elimination of present machinery is ever present.

B. DEFINITION OF THE PROBLEM

This paper discusses the design of a simulation model of the 360's Operating System in an MVT environment on a model 67-2 machine. When jobstream statistics are provided, the simulation model is designed to predict system performance with a degree of accuracy necessary for

planning and evaluation purposes. By the manipulation of certain parameters and/or blocks, the model may be made to conform to a variety of configurations and scheduling schemes.

No data concerning the actual jobstream has been collected at the school as of this time, except for a manual count of the number of jobs run on a given day. This lack of information prohibits the use of comparative experimentation to validate the model other than at the most fundamental level. Comments concerning this phase of model development will therefore be restricted to a discussion of the techniques which should be used in experimentation when job data becomes available.

Validation of this model was conducted by applying a pseudo jobstream, which is described in detail in Chapter III, to the model and conducting a parametric analysis of the important aspects of all modules. In addition, these results were compared with the author's observations of the actual processing of similar jobs on the 360.

The conclusions drawn will necessarily be concerned only with the viability of the developed model.

CHAPTER II

SYSTEM DESIGN

The System/360 consists of numerous hardware devices and an operating system. A thorough knowledge of both these areas was a prerequisite to the development of this simulation model. This information is available from numerous I.B.M. publications; those referenced are thought to be representative and give a broad enough background for an understanding of the model.

The configuration discussed here consists primarily of 2 central processing units (CPU's), 512K bytes of main storage, 1 drum and 8 disk storage devices, 4 tape units, 2 card readers, 2 printers, a card punch, 12 communication terminals, a graphical display unit, 2 graphical plotters, and associated channel control units. These are all interconnected through a switching device which allows many configurations including separate, simultaneous processing by the 2 CPU's (9). Although the hardware can be utilized to support multiprocessing, the MVT Operating System does not provide these capabilities; hence, only mono-processing is considered in the model. Five methods are provided for interruption of the CPU (program, supervisor call, external, I/O, and machine check).(8)

The operating system is responsible for satisfying valid user request through proper job, task, and data management (5). The flow of system responses and user requests is shown in Appendix A. Job management in MVT handles scheduling on a priority basis, interprets the user requirements, assigns devices, and initiates and terminates

each job step. Task management includes resource allocation, and task supervision. Data management concerns itself with label processing, retrieval of data sets, buffer management and scheduling, and data set access (7).

An MVT environment is one which allows several tasks to be simultaneously in core storage. Input readers, output writers, several user jobs, resident routines, and operator communications routines can all be in core at the same time, subject only to storage limitations. The various tasks compete on a priority basis for the CPU; interrupt handling procedures include switching to the highest ready task when appropriate. Thus simultaneous job processing and I/O operations are possible with this multitasking situation (10,11).

A job presented to the computer is thus read into the queue of jobs awaiting processing by a reader/interpreter task, executed by an initiator/terminator task, and has its output processed by one or more writer tasks. The possible advantages of multitasking in terms of efficient equipment utilization are apparent; however, the analysis of such a system seems only tractable by means of simulation techniques. Available analytical techniques (14) require simplifying assumptions unrealistic for modeling a complex system such as MVT.

CHAPTER III

I. MODEL DESIGN

A. GENERAL

It is possible to model every individual operation of a computer system including hardware, control program, and jobstream in an exact manner. Such a simulation, when applied to multiprogramming situations will result in a cumbersome model which requires as much or more running time as the actual operation. It is the objective of this study to provide a simple model whose behavior is representative of that of the real system. At the same time it is necessary to provide expansion points for future study and model refinement. Thus a complex operation may be represented by a simple delaying "advance" block (2); if it is necessary to examine that facility more closely, the single block could be replaced with a series of detail blocks which more accurately approximate the operation.

The ease with which various sub-systems could be constructed from block diagrams and the provisions for handling interrupts of transactions on a priority basis led to the choice of GPSS/360 for implementation of the model. Certain simplifications of the model were necessary to accommodate language restrictions or those imposed by the demands of processing time.

The model reflects the configuration of the IBM 360 model 67-2 computer operated at the U.S. Naval Postgraduate School under release 14 of the Operating System generated for option 4, Multiprogramming with a Variable number of tasks. It is expected that minor modifications of the model will be required for its application to subsequent

versions of the MVT operating systems.

B. JOBSTREAM GENERATOR

The jobstream generator module was designed for the purpose of exercising the model only; it must be replaced in order to obtain useful experimental results. Each jobstep is represented by a transaction which is created by a generate block. The job and step attributes are represented by the values of parameters associated with each transaction as indicated in Table 1.

Parameter	Description
PR	The job's PRTY parameter initially, and subsequently its dispatching priority during execution
P1	Varies; used for communication between modules
P2	Number of steps in the job
P3	Number of input records (cards)
P4	Number of Class A output records (print)
P5	Size of REGION required in K (1024 bytes)
P6	Execution time (CPU) in milli-seconds
P7	Mean CPU time between interrupts
P8	Number of Class B records (punch)
P12	Initiator being utilized during execution

Table 1. Parameter Association for Jobstream Generator

The present jobstream consists of identical one step jobs. The parameters were chosen to be representative of an average student job which exercised all modules of the model. The selected parameters are shown in Table 2.

Parameter	Value	Meaning
PR	8	Selection Priority
P2	1	Number of steps
P3	100	Number of input cards
P4	2500	Number of lines of print
P5	100	Region size
P6	30000	Execution time in milli-seconds
P7	100	Milli-seconds between interrupts
P8	10	Number of output cards

Table 2. Parameter Values for Pseudo Jobstream

The priority of eight is the present default priority at the school. Execution time, region size, input cards, and output lines and cards are about average for a student job. The time between interrupts was an arbitrary number used to test the interrupt mechanism.

C. READER/INTERPRETER

The reader/interpreter module enters the jobstream into the system jobqueue and input spools.. Those jobs which are marked as having job control language errors are processed in a "jobfail" mode. Default parameters are not provided by the reader for jobs as this was thought to be a simple external task and would degrade the operation of the module. Each job step transaction must therefore be presented to the reader with all required parameters supplied.

The module will operate multiple input streams if desired, each of which needs 42K of core. The starting and stopping of readers is controlled by the contents of the jobqueue. Processing time for each job

is dependent upon the number of steps and the number of cards read. To operate the model the reading rate was assumed to be 1000 cards per minute, the maximum speed of the 2540 card reader. Since the reader/interpreter does not always operate this rapidly, modification should be made when timing information is available. The formula used for the number of spooling tracks needed, $(P3/401+1)10$ is based on the 2311 disk drive's track capacity and the present default of 10 tracks per extent.

D. OUTPUT WRITER

The system output writers handle all final output associated with each job and the purging of the job from the simulation. Multiple output devices are supported; each requires 28K of core. The class B (punched) output is allowed to back-up to a given level before it is punched; and then it operates until the queue is empty. The class A (printed) output writer operates for the entire simulation run unless it is redefined between start control cards. A start-stop system similar to that devised for the punch could be applied to the printers if desired. Processing time required for the writer is dependent on the number of records produced by the job and the speed of the output device. The use of special characters which slow the processing time were not considered in this model. It is felt that this situation would best be handled by providing a separate class for this type of output. The output formulas $P4/27+1$ and $P8/42+1$ for class A and B outputs respectively are based on the track capacity of the disks. The print speed was set at 1200 lines per minute.

E. IPL

The IPL module conducts the Initial Program Load by defining the systems configuration parameters for the simulation run. The IPL transaction enters core in the amount of the nucleus/system queue area plus the link pack/master scheduler area. It starts the appropriate number of readers, initiators, and writers, then moves to the start-stop loops where it operates for the remainder of the simulation. This module is provided to allow simulation of system restart, as well as a convenient and realistic method for specifications of system parameters.

F. INITIATOR/TERMINATOR

The initiator/terminator provides the processing support for the jobstream. The job, once having entered the initiator, is processed by the ENQ routine. It was felt that the enqueueing of resources was not of particular significance at this installation since the majority of users enqueue only on sharable resources, such as the system disks, spools, and program libraries. The routine is branched to, however, in order to provide for expansion if necessary. The initiator then assumes the dispatching priority of the job and obtains core space (REGION) for it. The model does not ensure that this region is of contiguous core due to the characteristics of the 'storage' block in GPSS; this occasionally allows a job to be processed which would normally be delayed due to core fragmentation (2). Proper operator technique and job classification minimizes the possibility of core fragmentation on the real system; hence, this discrepancy is not considered to unduly affect the model.

Device allocation is handled by the allocate routine which assigns

devices to each job step which requires non-temporary data sets on direct access or tape storage. Experience thus far with the actual system operation¹ has shown that temporary storage space is not a problem with the amount of spooling space provided, hence it is not considered here. If the number of spools should be reduced, this problem might bear closer examination. Private disk pack allocation is handled along the lines of present day usage of such job libraries. Since the policy regarding the use of such disks may well change in the future, this portion of the model should be made to conform with the policy imposed. There are presently four spools and two private library drives in both the actual system and the model.

The transaction is next released to process until completion. The CPU facility is preempted on a priority basis and a copy transaction is split off and acts as an interrupt for that step during processing. When the execution time parameter has been reduced to zero, the step is considered completed and the termination process is entered in order that devices may be freed and region released. If there are further steps, they now enter the initiation phase at the device allocation level and are similarly processed. If the previous job step was abnormally terminated, as indicated by an initial value of zero for the execution parameter, the following steps are "flushed". The last job step is routed through the DEQ routine following termination, indicating

¹ The author was able to obtain several listings of the Volume Table of Contents (VTOC) on each of the spooling disk packs at different intervals including periods of peak spooling activity. At no time did these volumes exceed 50% of their combined capacity.

job termination and releasing the initiator to select a new job.

G. INTERRUPTS

Only those interrupts which require task switching, i.e., those which place the current task in a wait state, are considered. This was done because other types of interrupts are already included in the job execution time. To include these would require more simulation time, complicate the model, and add little to the results. At random times during the processing (the distribution is determined by the interrupt parameters of the job step) these interrupts preempt the CPU and place the current task in a wait state; this is followed by interrupt processing time and the initiation of parallel I/O processing. Upon completion of post-I/O processing, the task is placed in a wait state and allowed to compete for the CPU facility. The buffer capability of GPSS which restarts the scan of transactions to dispatch that one which possesses the highest priority is used for task switching purposes.

H. I/O

Although facilities have been provided for channels, control units, and device units, little I/O operation is simulated due to the time consuming complexity of the operation. These operations are represented rather well by simple advance blocks with times provided from standard distributions. If the area of I/O operation is found to be of interest and statistics from the actual system can be gathered, this module may be expanded to handle I/O with greater accuracy. The formulas in Table 3 are provided for computation of Channel and Control Unit facility given the unit facility number (F).

Actual	Model
Channel 0 (F=21...84)	8
Control Unit	$(F+111)/12$
Channel 1 (F=85)	9
Control Unit	17
Channel 2 (F=86...97)	10
Control Unit	$(F-14)/4$

Table 3. Channel and Control Unit Calculation Formulas

Individual unit assignments are listed in Appendix B.

As an example the model card punch is facility 47 (F=47). Since F is less than 85 the actual Channel is 0 (model facility 8). The Control unit is the least integer in $(47+111)/12=13$. These formulas should be used for rapid calculation in I/O operations.

I. PRIORITY

Transaction priorities are limited by GPSS to values between 0 and 127, whereas those in the actual system range from 0 to 255; therefore the model uses numbers equal to one half of those used in the actual system (rounded down to the nearest integer), to represent priorities.

CHAPTER IV

I. RESOLUTION

A. RESULTS

As there was no data available concerning jobstream characteristics, measurements on hardware utilization, or processor utilization by system tasks (overhead), the validating results are from model operations with the pseudo jobstream only. The runs were made with the parameters described in Chapter III and from one to three initiators. Statistics were gathered after three and six simulated hours. During the first hour, job generation only was done to provide an initial backlog. It was intended that the results be realistic and not necessarily accurate due to the complete lack of accurate input statistics. The results obtained are summarized in Table 4.

Statistic	1 Initiator		2 Initiators		3 Initiators	
	3 hrs	6 hrs	3 hrs	6 hrs	3 hrs	6 hrs
Jobs Generated	184	361	184	361	184	361
Jobs Read	71	71	71	168	113	184
Jobs Executed	21	54	42	108	63	162
Jobs Printed	21	54	42	108	63	162
Jobs Punched	21	42	42	105	63	147
Average CPU Utilization	.060	.077	.121	.151	.182	.227
Average Core Utilization	.039	.320	.439	.549	.561	.701
Average Spool Utilization	.055	.058	.065	.070	.056	.080

Table 4. Sample Results from Pseudo Jobstream Runs

As expected, the amount of work accomplished was related to the number of initiators in operation. Resource utilization also increased as expected. The CPU utilization is much lower than actual since the system overhead has not yet been incorporated. Intermittent punch and reader operation based on queue sizes account for the differences in these figures and the increase in the number of jobs read. Other statistical output from these runs and a run of twenty-four hour duration gave similar results. In each case this data supports the viability of the model.

The actual running time of the model was a function of the number of transactions and the number of blocks in the system, with the former being by far the greatest factor. When transactions were changed from their representation as a job step to that of a complete job, the time was decreased by more than a factor of three. Exact figures for model timing are not available due to the absence of an accounting routine on the present system. The use of link chains, where possible, was also found to improve efficiency. The elimination of blocks which added little to the model due to their infrequent use or infinitesimal contribution to the delaying time of a transaction enabled a much larger sample of jobs to be completed in this same amount of computer time.

B. RECOMMENDATIONS

When accurate operational data is made available, thorough experimentation must be conducted to completely validate the model. These experiments should show that the model is an accurate representation of the system in every area which is to be studied by the use of this

simulation model. A jobstream should be chosen which depicts a wide variety of jobs representative of those presented to the computer facility for processing. Data must then be gathered from actually running this jobstream and comparisons made with the statistical output of the GPSS simulation of the same jobstream. Direct comparisons should then be made of all queue sizes, task processing times, resource utilization, and total tasks processed. Care must be taken that jobs are run under the same configuration and parameter (number of initiators, readers, writers, etc.) circumstances and that observation intervals are the same for both system and model.

Provided these experimental results show the model to be valid, it may be utilized in several capacities. Individual parameters should be varied to determine optimum operational methods. New modules may be added to test the efficiency of adding new equipment. Configuration may be varied to determine optimal equipment mixes. The jobstream can be modified to find better means of job classification. Measurements can be made at points in the model which are unavailable or difficult to obtain from the system. In any case, best results will be obtained if all sections except the one being studied are left unchanged at first. Secondary experimentation to study interaction effects should then be undertaken.

C. CONCLUDING REMARKS

To be effective the model must be simple. Only those portions of the simulation which are to be studied should be expanded in detail. A bulky model results in clouded results and becomes as time consuming as the system itself.

The results will be no better than the statistical information upon which it is based. Knowledge of the actual jobstream is therefore of utmost importance, followed closely by machine timing information. With this data available for a base, sound predictions concerning new methods may be obtained.

CHAPTER V

SUMMARY

This paper reports on the design of a simulation model of the IBM 360 Computer operating in a Multi-programming with a Variable Number of Tasks (MVT) environment. The design is based on the operating system and computing equipment available at the U.S. Naval Postgraduate School's Computer Facility.

The model is written in the General Purpose Simulation System/360 (GPSS) language and is basically macroscopic in design. Each portion of the model can operate separately or be expanded in greater detail to allow for closer study of that particular aspect of operation.

A brief outline of the System/360 is given; included is a description of the hardware components, responsibilities of the operating system, elements of the MVT environment, and further system references.

There are seven major operational modules: a test jobstream generator; reader/interpreter; interruptor; and input/output. Also provided are an exponential function, commonly used variables, a clock, and a run control section.

Since jobstream, timing, and overhead data were not available from the facility, the model was operated using a pseudo jobstream for testing purposes. The results of the experiment were as anticipated and showed the model to be viable. It was also found that the efficiency of the model was enhanced by reducing the number of transactions to a minimum and by simplifying operations to only a few of

the more important and/or time consuming steps.

The full validation of the model requires the utilization of presently unavailable system and jobstream timing information; an outline of the required validation trials is included. Model utilization is described with the stipulation that the verifying simulation tests are favorable.

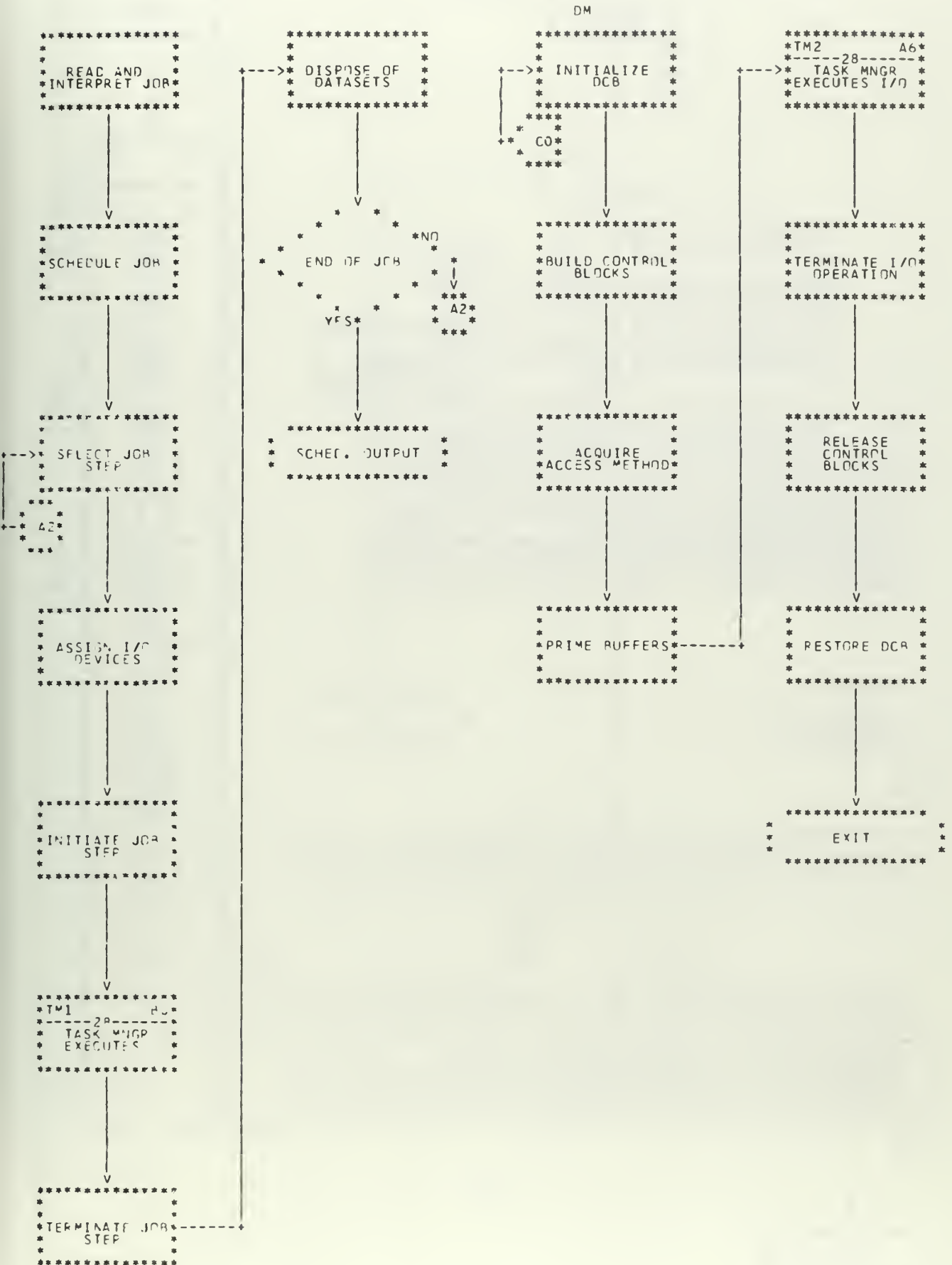
It is concluded that simple models operate more efficiently and that many system and jobstream statistics must be known for effective model operation.

BIBLIOGRAPHY

1. Havender, J.W., et. al., IBM Systems Journal, Vol 7, No. 2, pp. 73-144, International Business Machines Corporation, Armonk, New York, 1968.
2. I.B.M., General Purpose Simulation System/360 Introductory User's Manual, (H20-0304), International Business Machines Corporation, White Plains, New York, 1968.
3. I.B.M., General Purpose Simulation System/360 OS (360A-CS-17X) Operator's Manual, (H20-0311), International Business Machines Corp., White Plains, New York, 1968.
4. I.B.M., General Purpose Simulation System/360 User's Manual, (H20-0326), International Business Machines Corp., White Plains, New York, 1967.
5. I.B.M., IBM System/360 Operating System Concepts and Facilities (C28-6535-3), International Business Machines Corp., Poughkeepsie, New York, 1968.
6. I.B.M., IBM System/360 Operating System Storage Estimates (C28-6551-6), International Business Machines Corp., Poughkeepsie, New York, 1968.
7. I.B.M., IBM System/360 Operating System Supervisor and Data Management Services (C28-6646-1), International Business Machines Corp., Poughkeepsie, New York, 1968.
8. I.B.M., IBM System/360 Principles of Operation (A22-6821-6), International Business Machines Corp., Poughkeepsie, New York, 1967.
9. I.B.M., IBM System/360 System Summary (A22-6810-9), International Business Machines Corp., Poughkeepsie, New York, 1968.
10. I.B.M., OS/360 MVT Job Management-PLM (y28-6660-2), International Business Machines Corp., Poughkeepsie, New York, 1968.
11. I.B.M. OS/360 MVT Supervisor-PLM (Y28-6659-2), International Business Machines Corp., Kingston, New York, 1968.
12. Katz, Jesse H., "An Experimental Model of System/360", Communications of the A.G.M., Vol 10, No. 11, pp. 694-702, Nov. 1967.
13. Martin, James, Design of Real-Time Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967.
14. Wald, Bruce, The Throughput and Cost Effectiveness of Monoprogrammed Multiprocessing Digital Computers, Naval Research Lab., Washington, D.C., 1967.

JOB MANAGEMENT

DATA MANAGEMENT



USER PROGRAM



EQUATES

[illegible]

RDR80	EQU	69,F	READER ORO	TAPE GCO
TAPC0	EQU	81,F	SEVEN TRACK	TAPE OC1
TAPC1	EQU	82,F	NINE TRACK	TAPE OC2
TAPC2	EQU	83,F	NINE TRACK	TAPE OC3
TAPC3	EQU	84,F	NINE TRACK	TAPE OC3
DRM00	EQU	85,F	DRUM ICO	
DSK00	EQU	86,F	DISK DRIVE	200
DSK01	EQU	87,F	DISK DRIVE	201
DSK02	EQU	88,F	DISK DRIVE	202
DSK03	EQU	89,F	DISK DRIVE	203
CRT70	EQU	90,F	DISPLAY	270
DSK90	EQU	94,F	DISK DRIVE	290
DSK91	EQU	95,F	DISK DRIVE	291
DSK92	EQU	96,F	DISK DRIVE	292
DSK93	EQU	97,F	DISK DRIVE	293

FUNCTION DEFINITIONS				
EXP	FUNCTION	RN1,C24	AN EXPONENTIAL	FUNCTION
U	C	104	3	509
6	915	1.2	8	1.83
9	2.3	2.52	95	3.2
98	3.9	4.6	998	7
			999	9997
				8
				69
				2.12
				3.5

STORAGE DEFINITIONS				
CCRE	STORAGE	512	CORE CONFIGURATION	
INIT	STORAGE	2	NUMBER OF INITIATORS	
JOBQ	STORAGE	60	MAXIMUM SIZE OF JOB QUEUE	
PDSK	STORAGE	2	NUMBER OF DISKS FOR JOBLIBS	
RDR	STORAGE	1	NUMBER OF INPUT READERS	
SPOOL	STORAGE	8000	NUMBER OF TRACKS FOR SPOOLING	
TAPE7	STORAGE	1	NUMBER OF SEVEN TRACK TAPES	
TAPE9	STORAGE	3	NUMBER OF NINE TRACK TAPES	
WTR	STORAGE	2	NUMBER OF CLASS A OUTPUT WRITERS	

VARIABLE DEFINITIONS				
CHAP1	VARIABLE	P1*8+5	SELECT TO DISPATCH PRIORITY	
CHAP2	VARIABLE	(PR-5)/8	DISPATCH TO SELECT PRIORITY	
HOURL	VARIABLE	360C000	ONE HOUR IN MILLI-SECONDS	
MIN	VARIABLE	60C00	ONE MINUTE IN MILLI-SECONDS	
SEC	VARIABLE	10C0	ONE SECOND IN MILLI-SECONDS	
READ	VARIABLE	P3*6C	TIME TO READ CARDS	
SYSK	VARIABLE	136	NUC + SQA + LPA + MASTER SCHED.	
SYSDIN	VARIABLE	(P3/4C1+1)1C	NUMBER OF INPUT SPOOLING TRACKS	
SOUT	VARIABLE	P4/27+1+P8/42+1	NUMBER OF OUTPUT TRKS NEEDED	
SOUTA	VARIABLE	P4/27+1	NUMBER OF CLASS A OUTPUT TRKS	

```

SOUTB VARIABLE P8/42+1      NUMBER OF CLASS B OUTPUT TRKS
WRITE VARIABLE P4*50        TIME TO PRINT
*
*
*
*
IPL SEQUENCE INCLUDING DELAY (THIRD PARAMETER) TO BUILD UP JOB
QUEUE BEFORE STARTING PROCESSING.

GENERATE      ,V$HOUR,1,127
ENTER        CORE,V$SYSK
ADVANCE
SPLIT        1,SPUN
LENGTH OF TIME TO IPL

*
*
*
LOOP TO START AND STOP READER AT APPROPRIATE TIME

TEST L        S$JOBQ,10
ENTER        CORE,42
SEIZE        RDR31
ADVANCE
LOGIC S
GATE SF
LOGIC R
ADVANCE
RELEASE
LEAVE
TRANSFER
TIME TO INITIALIZE READER

SRDR
RDR31
JC8Q
RDR31
RDR31
CCRE,42
,SRDR
TIME TO TERMINATE READER

*
*
*
LOOP TO START AND STOP PUNCH AT APPROPRIATE TIME

TEST G        Q$PUN,20
ENTER        CORE,28
SEIZE        PUN32
LOGIC S
TEST E        Q$PUN,0
LOGIC R
RELEASE
LEAVE
TRANSFER
SOUT B QUEUE SIZE TO START PUNCH
REGION SIZE OF OUTPUT WRITER

SPUN
PUN32
Q$PUN,0
PUN32
CORE,28
,SPUN

*
*
*
CLOCK WHICH DETERMINES THE LENGTH OF THE SIMULATION RUN

GENERATE V$HOUR
TERMINATE 1

*
*
*
*
JOB GENERATOR IS USED FOR TESTING PURPOSES ONLY

GENERATE      60000,FN$EXP,,124      JOB GENERATOR (TEST)
ASSIGN        2,1                    NUMBER OF STEPS
ASSIGN        3,100                  NUMBER OF CARDS
ASSIGN        4,2500                 NUMBER OF LINES OF PRINT

```

REGION SIZE
EXECUTION TIME (CPU)
MEAN TIME BETWEEN INTERRUPTS
NUMBER OF CARDS PUNCHED

DEFAULT PRIORITY

TIME TO INITIATE JOB

TIME TO END

DID JOB FAIL

```

5,100
6,30000
7,100
8,10
**
READER/INTERPRETER
GATE LS
ENTER RDR31
ENTER RDR
ENTER JCBO
ASSIGN SPCOL,V$SYSIN
ADVANCE 1,8
LEAVE V$READ
PRIORITY RDR
PRIORITY P1,BUFFER
**
INITIATOR/TERMINATOR
ENTER INIT
PRIORITY 125
ENTER CCRC,58
LOCATE A FREE INITIATOR
TRANSFER ALL,1, LAST,3
SEIZE INIT1
ASSIGN 12,INIT1
SEIZE INIT2
ASSIGN 12,INIT2
TRANSFER INIT3
SEIZE 12,INIT3
ASSIGN INIT4
TRANSFER INIT4
SEIZE 12,INIT4
ASSIGN INIT5
TRANSFER INIT5
SEIZE 12,INIT5
LAST
**
JSTRT PREEMPT CPU,PR
ADVANCE
PRIORITY
LEAVE
ENQ ADVANCE
**
SSTRT RETURN
ALLOC ENTER
TEST G

```

PREEMPT	CPU,PR	TIME FOR ALLOCATION
ADVANCE	1,SVC	
SPLIT	P6	
ADVANCE	SPNOL,V\$SYSIN	
LEAVE		
ADVANCE	CORE,P5	TIME TO TERMINATE JOB STEP
LEAVE	2,SSRT	
LOOP		
ADVANCE	CPU	TIME TO TERMINATE JOB
RETURN	V\$CHAP2	
PRIORITY	P12	
RELEASE	INIT	
LEAVE	1,PUNCH	
SPLIT		

```

CLASS A OUTPUT WRITER

ENTER
ADVANCE
LEAVE
LEAVE
LEAVE
TERMINATE

WTR
V$WRITE
SPCOL,V$SOUTA
JOBQ
WTR

```

```

*
*
*
*
PUNCH
  QUEUE
  GATE LS
  SEIZE
  ADVANCE
  LEAVE
  RELEASE
  DEPART
  TERMINATE
CLASS B OUTPUT WRITER
  PUN
  PUN
  PUN
  SPOGL,V$SOUTR
  PUN
  PUN

```

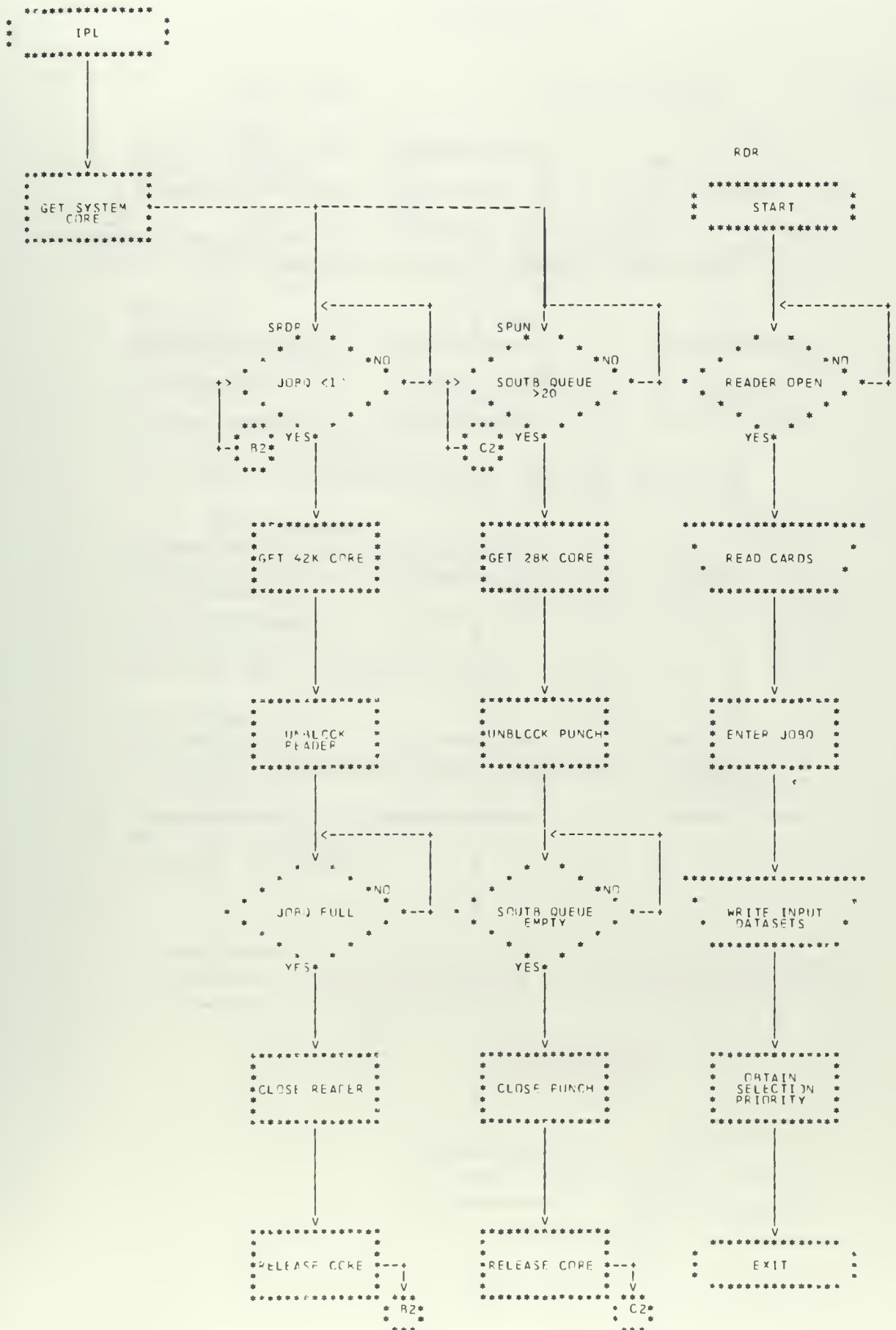
* * * *	SVC INTERRUPTER		TIME BEFORE INTERRUPT
SVC	ADVANCE PRIORITY PREEMPT ADVANCE RETURN TERMINATE	P7 125 CPU, PR, WAIT, 6, RE CPU	PROCESSING TIME FOR SVC
* * * *	WAIT STATE AREA		
WAIT	ADVANCE PREEMPT	100C CPU, PR	WAIT TIME


```

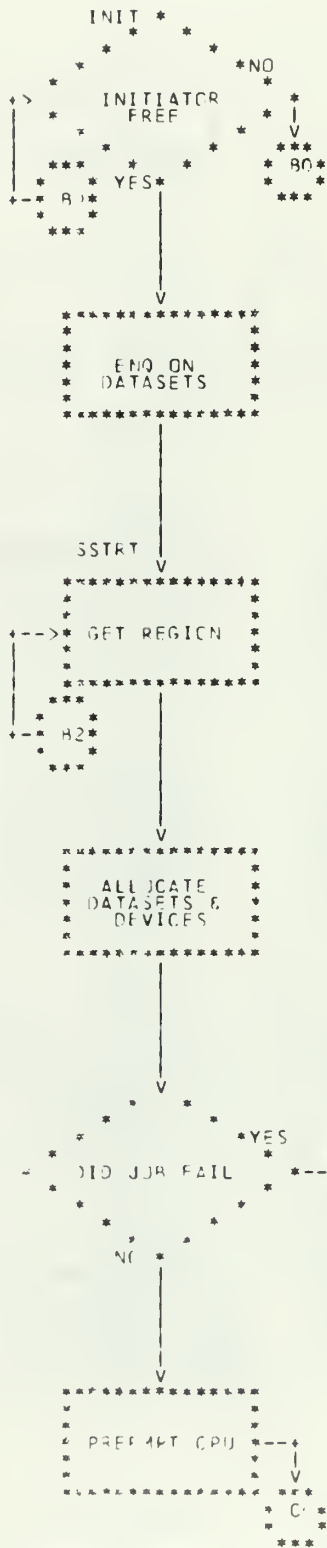
* * * * *
TRANSFER      ,BACK
CONTROL SECTION FOR SIMULATION RUN
FIRST PARAMETER ON START CARDS IS NUMBER OF HOURS TO BE SIMULATED

SIMULATE      6,,3,1
START
END

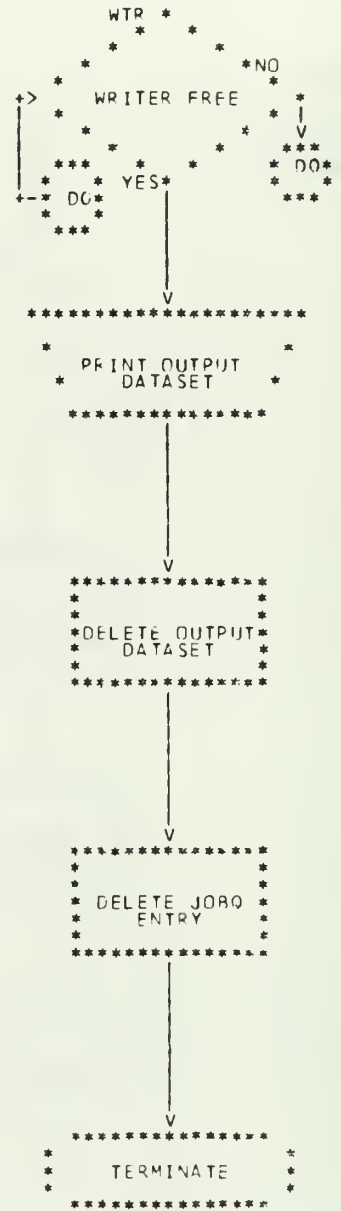
```



INITIATOR/TERMINATOR



OUTPUT WRITERS



INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2. Library U.S. Naval Postgraduate School, Monterey, California	2
3. Sponsor Department of the Army ATTN: OPD (Sig C) Washington, D. C.	1
4. Mr. G.L. Barksdale, Jr. Department of Mathematics U.S. Naval Postgraduate School, Monterey, California	1
5. Prof. George Heidorn Department of Operations Analysis U.S. Naval Postgraduate School, Monterey, California	1
6. Cpt. A.K. Varnell 1560 Hollingsworth Drive Mountain View, California 94040	1
7. LCDR Roger Bacon Box 1464 U.S. Naval Postgraduate School, Monterey, California	1
8. Department of Operations Analysis Naval Postgraduate School Monterey, California	1

INTERNALLY DISTRIBUTED
REPORT

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP	
3. REPORT TITLE A Simulation Model of the IBM 360 Computer in an MVT Environment			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Masters Thesis March 1969			
5. AUTHOR(S) (Last name, first name, initial) VARNELL, Allan K.			
6. REPORT DATE April 1969		7a. TOTAL NO. OF PAGES 38	7b. NO. OF REFS 14
8a. CONTRACT OR GRANT NO. N/A		8a. ORIGINATOR'S REPORT NUMBER(S) N/A	
b. PROJECT NO. N/A			
c.		8b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		N/A	
10. AVAILABILITY/LIMITATION NOTICES This document has been approved for public release and sale; its distribution is unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT This paper reports on the design of a computer simulation model written in GPSS. A brief description of the System/360 operating system is given. The model consists of macroscopic modules representing distinguishable computer tasks which are capable of independent operation and/or more detailed expansion. A pseudo jobstream of sufficient detail was used to test the viability of this model. Guidelines for experimentation (which was prohibited by a complete lack of data) are outlined; suggested uses of the model are given.			

Security Classification

14

KEY WORDS

LINK A

LINK B

LINK C

ROLE

WT

ROLE

WT

ROLE

WT

Computer Simulation

thesV327

A simulation model of the IBM 360 comput



3 2768 001 01358 4

DUDLEY KNOX LIBRARY